

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

---

## **M2M senzorový modul**

M2M sensor module

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání bakalářské práce

Student: **Marek Darmo, DiS.**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R059 Mobilní technologie

Téma: M2M senzorový modul  
M2M Sensor Module

Zásady pro vypracování:

Navrhňte ústřednu pro sběr údajů ze sítě senzorů komunikujících pomocí sběrnice 1-Wire. Základem bude bezdrátový modul Siemens XT75 a na něm běžící aplikace v J2ME. Data budou přenášena pomocí GPRS spojení na server ve formátu XML. Informace budou uživatelům zpřístupněna pomocí Web rozhraní.

1. Vytvořte vhodné rozhraní 1-Wire / XT75.
2. Implementujte aplikaci pro obsluhu čidel a komunikaci se serverem.
3. Server přijímající XML zprávy.
4. WWW rozhraní pro zpřístupnění měřených hodnot.

Aplikaci otestujte s minimálně třemi připojenými senzory.

Seznam doporučené odborné literatury:  
<http://www.cinterion.com/>  
Kim Topley, J2ME in a Nutshell (O'Reilly Java)

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michal Krumník**

Datum zadání: 30.11.2008  
Datum odevzdání: 07.05.2010

doc. Dr.Ing. Eduard Sojka  
*vedoucí katedry*

prof. Ing. Ivo Vondrák, CSc.  
*děkan fakulty*

### **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7.5.2010

.....

## **Poděkování**

Rád bych na tomto místě poděkoval všem, kteří mi pomáhali při tvorbě této práce, ať už přímo či nepřímo. Zvláště pak děkuji panu Ing. Michalu Krumníkovi, vedoucímu mé bakalářské práce, za jeho trpělivost, cenné rady a podnětné připomínky.

## **Anotace**

Cílem práce je využití mobilních technologií prostřednictvím M2M GSM modulu k industriálnímu sběru dat na vzdáleném místě a jejich přenosu z tohoto místa do centrálního datového centra. Zdrojem dat může být jakákoliv elektronicky měřitelná veličina nebo stav. Práce předkládá jednu možnou realizaci takového zařízení s GSM modulem firmy Cinterion a třemi teplotními senzory na společné datové 1-Wire sběrnici.

**Klíčová slova:** M2M, GSM, 1-Wire, DS18S20, Cinterion XT75, TC65, J2ME

## **Anotation**

Purpose of this work is take advantage of mobile technology by M2M GSM module for industrial data collection at distant place and their transfer from this place to central data center. Source of data can be any electronically measurable value or state. Labour presents one possible realization of such device with GSM module from Cinterion company with three temperature sensors shared on 1-Wire data bus.

**Keywords:** M2M, GSM, 1-Wire, DS18S20, Cinterion XT75, TC65, J2ME

## Seznam použitých symbolů a zkratk

M2M	- machine-to-machine
XT75	- GSM modul Cinterion
I <sup>2</sup> C	- Inter-Integrated Circuit, multi-masterová sériová sběrnice
1-Wire	- komunikační sběrnice vyvinutá firmou Dallas pro přenos dat po jedné lince
GSM	- Global System for Mobile Communications
XML	- Extensible Markup Language
CSV	- Comma-separated Value
GPRS	- General Packet Radio Service
J2ME	- Java 2 Platform Micro Edition ?
IMP-NG	- Information Module Profile No Gui. JSR-228
FLASH	- typ bezpříkonové paměť EEPROM, zapisovatelná ve velkých blocích
PC	- Personal Computer, osobní počítač
HTTP	- Hypertext Transfer Protocol
POST	- HTTP metoda pro posílání dat z formuláře na server
PCB	- Printed Circuit Board, deska plošných spojů
SIM	- Subscriber Identity Module
SMS, MMS	- Short Message Service, Multimedia Message Service
GPIO	- General Purpose Input Output pins, vstupně výstupní linky modulu XT75
SCL, SDA	- Serial Clock Line, Serial Data Line
Tx, Rx	- Transmit, vyslat, Receive, přijmout
Maxim	- Maxim Integrated Products, Inc.
Dallas	- Dallas Semiconductor Corp., v roce 2001 plně zakoupený firmou Maxim
Cinterion	- Cinterion Wireless Modules je nová společnost, která byla založena v červnu 2008 osamostatněním divize s moduly společnosti Siemens Wireless Modules
master zařízení	- zařízení nebo integrovaný obvod, který má v definované komunikaci postavení řídicího člena
slave zařízení	- zařízení nebo integrovaný obvod, který má v definované komunikaci postavení podřízeného člena
reset pulse	- posloupnost logické úrovně napětí na sběrnici, kterou master zařízení definuje počátek komunikace
presence pulse	- posloupnost logické úrovně napětí na sběrnici, kterou slave zařízení odpovídá na reset pulse
pullup	- rezistor nebo jiný obvod zajišťující napěťovou úroveň, v zapojení s otevřeným kolektorem, pokud není tranzistor sepnut.
datasheet	- listina prokazující vlastnosti a specifikaci obvodu či zařízení dané výrobcem

## Obsah

1.	Úvod	1
2.	Popis M2M technologie	2
3.	Modul Cinterion XT75	3
4.	Blokové schéma zařízení	4
5.	Funkce zařízení	5
5.1.	Příklad funkce.....	5
6.	Teorie hardwarových částí	7
6.1.	Sběrnice 1-Wire.....	7
6.2.	Sběrnice I <sup>2</sup> C.....	10
6.3.	Teplotní čidlo DS18S20.....	12
6.4.	Hardwarový most DS2482-100.....	17
7.	Hardwarový návrh	23
7.1.	Schéma zapojení modulu DS2480-100.....	23
7.2.	Stanovení pullup rezistoru I <sup>2</sup> C.....	24
7.3.	Návrh a výroba PCB.....	24
7.4.	Popis vývojového kitu M2M Board.....	25
7.5.	Realizace I <sup>2</sup> C na modulu XT75 přes linky GPIO.....	26
8.	Softwarový návrh	27
8.1.	Popis vývojového prostředí.....	27
8.2.	Java ME.....	28
8.3.	Popis software modulu XT75.....	30
8.4.	Získání atomového času J2ME/PHP.....	35
8.5.	XML Server.....	36
9.	WWW rozhraní pro zpřístupnění naměřených hodnot	37
10.	Reálný provoz zařízení	39
11.	Závěr	41
12.	Literatura	43

## 1. Úvod

Téma této bakalářské práce vzniklo jako vlastní téma a myšlenka praktického zhodnocení získaných informací při studiu oboru Mobilních technologií studijního programu Informačních a komunikačních technologií při katedře Fakulty elektrotechniky a informatiky Vysoké Školy Báňské. Hlavní myšlenkou celé práce je využití „mobility“ mobilních technologií také mimo standardní hlasové a datové služby poskytované zejména koncovým uživatelům, jako je přenos řeči nebo SMS či MMS. Ve výběru tématu jsem se přiklonil spíše k industriálnímu využití mobilních zařízení a to ke sběru a přenosu dat ze vzdáleného místa do datového centra, přičemž zdrojem dat může být jakákoliv elektronicky měřitelná veličina nebo stav. Pro realizaci tohoto modelového případu jsem po konzultaci s katedrou a vedoucím práce vhodně využil M2M modul XT75 od firmy Cinterion (bývalý název Siemens), který mi byl katedrou zapůjčen. Tento datový modul zabezpečuje logiku pro sběr dat a jejich vzdálený přenos do datového centra. Součástí modulu nejsou žádná čidla, která by poskytovala různé měřené veličiny a proto bylo třeba zvolit dodatečné hardwarové rozšíření. Kladl jsem důraz na univerzálnosti a jednoduchosti celého řešení a snadné připojitelnosti k modulu sbírajícímu data. Za vhodnou koncepci jsem po zvážení všech předností zvolil 1-Wire sběrnici od firmy Dallas Semiconductor Corp. (nyní Maxim Integrated Products, Inc.), která v rozsahu desítek metrů umožňuje snadno a jednoduše připojit různé hardwarové prvky pro sběr dat. Pro názornost a prezentaci jsem zvolil tři teplotní 1-Wire senzory DS18S20 a v mém případě realizující funkci měření venkovní teploty, vnitřní teploty a teploty vytápěcího zařízení.

Ačkoliv se zadání bakalářské práce na první pohled zdálo jako běžný úkol, nakonec bylo nutné zpracovat spoustu nových informací a kroků. Obsah této práce lze ve zkratce shrnout takto:

- nastudování teoretického základu použitých integrovaných obvodů, sběrnic a modulů
- návrh zapojení do funkčního celku a výroba prototypu plošného spoje
- nastudování zvláštností jako synchronizace času, softwarový I<sup>2</sup>C protokol, protokol pro ovládání hardwarového mostu
- napsání ovládacího software použitého modulu v J2ME, ladění a testování
- napsání software serveru přijímajícího XML zprávy a zobrazení na www
- reálný provoz zařízení



## **2. Popis M2M technologie**

M2M technologie zahrnuje využití různých typů zařízení, která jsou schopna vzájemně navázat datovou komunikaci a vyměnit si požadované informace. Zkratka M2M vyjadřuje tuto vzájemnou výměnu jako Machines-to-Machines. V řeči operátorů mobilních služeb se častěji používá Mobile-to-Mobile. M2M rozhraní umožňuje monitorovat a manipulovat se vzdálenými zařízeními. Základní struktura M2M technologie zahrnuje centrální systém, který je schopen spojení se systémy ve vzdálených lokalitách. Toto spojení umožňuje centrálnímu systému sbírat data ze vzdálených zdrojů nebo jim předat data ke zpracování.

Některé příklady využití M2M technologií:

- zavlažovací a kanalizační systém, městské semaforey a dopravní systém
- pasti na škůdce, řízení domácích spotřebičů, řízení závory, zabezpečení domu
- zabezpečení automobilů, jejich údržba a monitoring
- systém pro odstavné parkoviště, pacienty, hendikepované
- řízení uzávěrů a klapek, monitoring chladíren, výdejních automatů
- vzdálený monitoring spotřeby elektrické energie, plynu apod.

### 3. Modul Cinterion XT75

Modul Cinterion XT75 patří do rodiny M2M modulů, kde vývojáři mohou implementovat sledovací funkce díky vestavěné GPS technologii. Tato funkce zde sice není využita, nicméně modul navíc obsahuje výkonnou Java platformu, která umožňuje vyvinout komplexní aplikace ve velmi krátkém čase. Navíc díky Quad-Band pásmu je schopen modul pracovat kdekoli na světě. XT75 je vybaven také technologií EDGE pro nejvyšší rychlost přenosu dat. Nejdůležitější vlastnosti modulu XT75 jsou shrnuty v tabulce 3-1.

Obsah
<b>Obecné vlastnosti XT75:</b>
State-of-the-art GPS receiver (16 channels)
Quad-Band GSM (850/900/1800/1900 MHz)
EDGE (E-GPRS) multi-slot class 10 (XT75)
GPRS multi-slot class 12
Java™ IMP-NG
Integrated TCP/IP stack
RIL driver for Microsoft® Windows Mobile™ based devices
RLS Monitoring for Jamming detection
Paměťové místo pro Java programy: Flash File System 1200kB a RAM 400kB

*Tabulka 3-1: Vlastnosti modulu Cinterion XT75.*

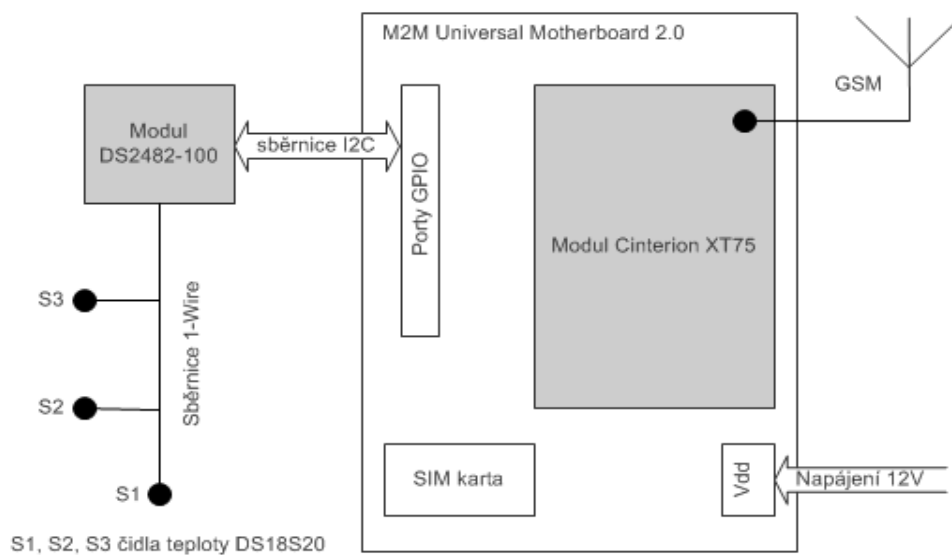
Aplikační kód a data aplikace sdílejí stejné místo v paměti Flash a RAM. Java aplikace může zpřístupnit přídavné hardwarové vybavení modulu:

- až deset digitálních I/O linek například jako výstup pro stavové led nebo vstup tiskového tlačítka
- jeden I<sup>2</sup>C/SPI Interface
- jeden D/A a dva A/D převodníky
- sériový port (RS-232 API)

#### 4. Blokové schéma zařízení

Celé pojetí bakalářské práce je ve smyslu návrhu pomocí vývojových prostředků, nikoliv realizace a nasazení pro koncového uživatele. Pro návrh je použita pomocná vývojová deska pro GSM moduly, která má vyvedeny veškeré sběrnice těchto modulů. Dále je vybavena potřebnými obvody pro napájení a obsahuje slot na SIM kartu.

Tři teplotní čidla jsou napojena na společnou 1-Wire sběrnici a protože modul XT75 není vybaven podporou pro připojení sběrnice 1-Wire, ale obsahuje port sběrnice I<sup>2</sup>C, je připojení teplotních čidel k modulu realizováno přes hardwarový most DS2482-100. Hardwarový most realizuje převod protokolu I<sup>2</sup>C na protokol sběrnice 1-Wire a naopak. Návrh zapojení hardwarového mostu je součástí této práce. Na obrázku 4.1 je blokové schéma zařízení.



Obr. 4.1: Blokové schéma zařízení.

## 5. Funkce zařízení

Úkolem navrhovaného zařízení je změřit teplotu ve třech místech a v pravidelných časových intervalech odeslat tyto data do datového centra na vzdáleném serveru připojenému k internetu. Jádrem navrhovaného zařízení je M2M modul, který je tzv. „prodlouženou rukou“ datového centra, čili modulem pracujícím ve vzdáleně lokalitě a sbírajícím data. Vybraný M2M modul, který sbírá teploty a zasílá je na vzdálený server, je GSM modul Cinterion XT75 s podporou programovacího jazyka Java J2ME v profilu IMP-NG.

Po svém startu modul XT75 synchronizuje svůj interní čas s atomovým časem pomocí GPRS připojení k internetu. V dalším kroku čeká na celou hodinu a následně začne s odečítáním hodnot na jednotlivých teplotních čidlech. Po odečtení teplot si hodnoty uloží do interní FLASH paměti a čeká na další celou hodinu. Pokud modul dojde k času 23:00 hod, provede poslední odečítání teplot a následně odešle všechny hodnoty na XML server přes GPRS spojení. Vždy po 23 hodině tedy modul posílá 24 hodnot měřených vždy v celou hodinu za aktuální den. Měření v následujícím dni začíná hodinou 00:00.

### 5.1. Příklad funkce

V příkladu je popsána situace, kdy byl modul spuštěn před 23 hodinou. V celou hodinu změřil teploty a odeslal ke zpracování na XML server. Zařízení je naprogramováno tak, aby současně při provádění programu zasílalo informace o stavu provádění programu a jeho činnosti na port RS232. Výpisy níže jsou tedy obsahem hyperterminálu spuštěného na vývojovém PC.

Nejdříve je provedena kontrola přítomnosti hardwarového mostu DS2482-100. Následuje získání přesného času z internetu a dále se měří teplota na třech čidlech, viz. tabulka 5-1.

```
Device DS2482 succesfully detected.  
AT+CCLK?  
+CCLK: "10/02/18,22:43:00"  
OK  
System time: Tue Jan 01 00:00:40 UTC 2002  
Atom time: Thu Feb 18 22:43:29 UTC 2010  
Time offset: 256689768969 in milliseconds  
XT75 time synchronized.  
Date: 18.02.2010  
Time: 23:00  
Inside sensor no.1 temperature reading...  
Temperature readed 21 C  
Heater sensor no.2 temperature reading...  
Temperature readed 35 C  
Outside sensor no.3 temperature reading...  
Temperature readed 11 C
```

*Tabulka 5-1: Start modulu a měření teploty.*

Naměřené teploty se odešlou ve formátu XML na server. Pro kontrolu je vypsán obsah XML souboru a obsah alternativního CSV souboru, viz. tabulka 5-2.

```
Temperature XML file sended.  
XML  
<?xml version="1.0" encoding="UTF-8"?>  
<message>  
<date>18.02.2010</date>  
<s1descr>Inside</s1descr>  
<s2descr>Heater</s2descr>  
<s3descr>Outside</s3descr>  
<sample>  
<h>23:00</h>  
<s1>21</s1>  
<s2>35</s2>  
<s3>11</s3>  
</sample>  
</message>  
  
CSV  
18.02.2010,Inside,Heater,Outside  
23,21,35,11
```

*Tabulka 5-2: Odeslání teplot na server.*

Na serveru se přijatý soubor XML zpracuje a zasláná teplotní data se připraví pro zobrazení na webové stránce. Obsahem stránky je pak graf zobrazující naměřené teploty v průběhu dne. Na obrázku 5.1 jsou zobrazeny teploty změřené a odeslané na server ve 23:00 hodin.



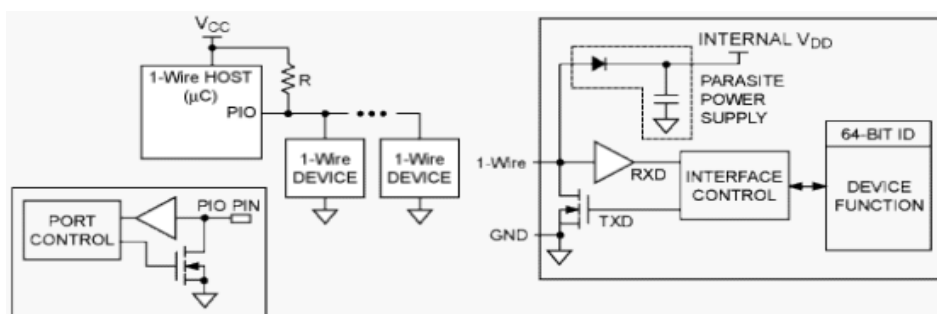
*Obr. 5.1: Teploty v průběhu vybraného dne.*

## 6. Teorie hardwarových částí

V této kapitole se nejprve zaměříme na použité komunikační sběrnice a následně na integrované obvody s těmito sběrnici pracující.

### 6.1. Sběrnice 1-Wire

Sběrnice 1-Wire byla vyvinutá firmou Dallas. Tato sběrnice představuje jednoduché signální schéma, které poskytuje dvoucestnou komunikaci mezi jedním master zařízením a periferními slave zařízeními přes jednodrátové spojení. 1-Wire je určena pro komunikaci v nízkých rychlostech a zajišťující přenos napájení a datových signálů po jednom vodiči, viz. obrázek 6.1. Takto byl definován i její název 1-Wire (jednodrátová). Ve skutečnosti tedy sběrnice potřebuje ke své činnosti dva vodiče, jeden zemnicí a druhý pro napájení a data. 1-Wire sběrnice je v koncepci podobná sběrnici I<sup>2</sup>C, poskytuje ale nižší rychlosti na delší vzdálenost. 1-Wire sběrnice byla původně konstruována pro menší vzdálenosti, postupně se ale její vývoj zdokonalil natolik, že dnes lze běžně dosáhnout vzdálenost desítek metrů, viz. podrobnější specifikace v literatuře [1].



Obr. 6.1: Komunikace po jednom vodiči.

Každé slave zařízení na sběrnici 1-Wire má výrobcem předprogramováno 64-bitové sériové číslo, které je jedinečné a je uloženo v paměti ROM (Read-Only-Memory). Tím je možno jednotlivé slave zařízení individuálně vybrat mezi ostatními zařízeními sdílejícími stejnou sběrnici. Pomocí sériového 64-bit ROM čísla lze tedy adresovat konkrétní slave zařízení a začít s ním komunikovat.

První část komunikace na sběrnici 1-Wire začíná vysláním signálu reset, který celou sběrnici sesynchronizuje. Signál reset vysílá pouze master zařízení. Následně se vybírá slave zařízení pro nadcházející komunikaci. Lze vybrat všechny slave zařízení, lze vybrat jedno konkrétní nebo lze oslovit následující zařízení na sběrnici pomocí binárního vyhledávacího algoritmu.

Tyto příkazy se označují jako síťové funkce nebo ROM příkazy, protože pracují s 64-bitovým ROM číslem a jejich funkcí je výběr slave zařízení k nadcházející komunikaci. Po tom co bylo vybráno cílové slave zařízení, všechny ostatní zařízení na sběrnici se nebudou nadcházející komunikace účastnit, dokud nebude master zařízením vyslán nový signál reset na sběrnici 1-Wire.

Jakmile je vybráno konkrétní slave zařízení, master zařízení může pokračovat v komunikaci a může začít posílat příkazy specifické pro dané slave zařízení. Může mu zasílat data nebo může data z tohoto slave zařízení číst. Každé slave zařízení poskytuje různé funkce, každý typ má svůj vlastní komunikační protokol, respektive konkrétní příkazy a specifický sled pro jejich vykonání. Nicméně všechny slave zařízení musí podporovat společný komunikační protokol pro proces jejich výběru. Všechny tedy reagují na příkaz reset a příkazy ROM.

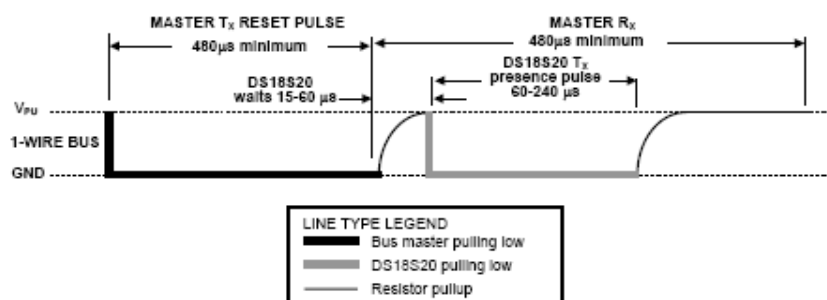
#### 6.1.1 1-Wire časování

Pro zachování integrity dat se na sběrnici 1-Wire používá přísný komunikační protokol. Je zde definováno několik typů signálů jako reset pulse, presence pulse, write 0, write 1, read 0, a read 1. Všechny tyto signály kromě presence pulse, jsou inicializovány pouze master zařízením.

#### 6.1.2 Inicializace sběrnice, reset a presence pulse

Jakákoliv komunikace na sběrnici 1-Wire začíná inicializační sekvencí, kdy master zařízení vyšle na sběrnici signál reset a pokud je připojeno nějaké slave zařízení, odpoví toto signálem presence pulse. Tato procedura poskytuje synchronizaci na 1-Wire sběrnici, a také informuje master zařízení o přítomnosti a připravenosti jednoho nebo více slave zařízení.

Tato procedura je popsána na obrázku 6.2., kde během inicializační sekvence master zařízení aktivuje sběrnici do nízké úrovně (logické nuly) na minimálně 480 $\mu$ s. Potom master zařízení sběrnici uvolní a přejde do režimu příjmu. Tento signál se nazývá reset pulse. Jakmile je sběrnice uvolněna, 5k $\Omega$  pullup rezistor aktivuje sběrnici zpět do vysoké úrovně (logické jedničky). Slave zařízení detekuje tento nárůst úrovně, počká 15 $\mu$ s až 60 $\mu$ s a potom aktivuje sběrnici do nízké úrovně na dobu 60 $\mu$ s až 240 $\mu$ s, což je nazýváno jako presence pulse. Detekcí presence pulse master zařízení dostane informaci o připojeném slave zařízení.



Obr. 6.2: Inicializační sekvence 1-Wire sběrnice.

Ostatním časováním signálu na sběrnici 1-Wire (write 0, write 1, read 0, read 1) se nebudeme detailněji zabývat, neboť v této práci se bude o správné časování na sběrnici 1-Wire starat k tomu určený hardwarový most DS2482-100. Detailnější specifikaci časování a průběhu signálů na sběrnici 1-Wire lze nalézt v literatuře [2].

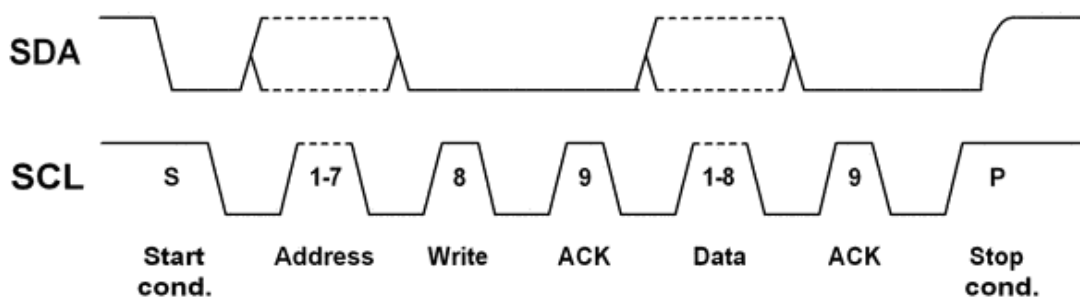


## 6.2. Sběrnice I<sup>2</sup>C

Sběrnice I<sup>2</sup>C je více masterová sběrnice. Je tvořena dvěma linkami, jednou datovou SDA a druhou synchronizační SCL. Obě linky jsou spínány tranzistorem s otevřeným kolektorem, proto je potřeba obě linky připojit přes pullup rezistor na kladné napájení a zajistit tak na linkách vysokou úroveň napětí pokud na sběrnici neprobíhá žádná komunikace. Master zařízení inicializuje veškerou komunikaci na sběrnici a dodává hodinové impulzy. Maximální délka sběrnice je dána jednak nejvyšší přípustnou kapacitou 400pF a jednak komunikační rychlostí na lince, která může být standardních 100kHz nebo v rychlejší variantě 400kHz. Každé slave zařízení musí mít svoji adresu, na kterou se master odkazuje při zahájení komunikace prvním bajtem. Tento první bajt je složen ze 7-mi bitové adresy slave zařízení a nejnižšího bitu, kterým master označuje požadavek na zápis (bit0=0) nebo čtení (bit0=1).

I<sup>2</sup>C komunikace může být zahájena pouze v případě, že sběrnice je volná. Master zařízení generuje časování SCL, řídí přístup ke sběrnici, generuje podmínky komunikace START a STOP a vymezuje počet datových bajtů přenášených mezi podmínkami START a STOP. Data jsou přenášena nejvýznamnějším bitem napřed. Po každém přeneseném bajtu následuje potvrzovací signál ACK od slave zařízení.

Komunikační události I<sup>2</sup>C sběrnice jsou uvedeny v tabulce 6-1. Ve výchozím stavu je sběrnice neaktivní a SDA i SCL linka se nachází ve vysoké úrovni napětí. Příklad časování je zobrazen na obrázku 6.3.



*Obr. 6.3: Příklad časování na sběrnici I<sup>2</sup>C.*

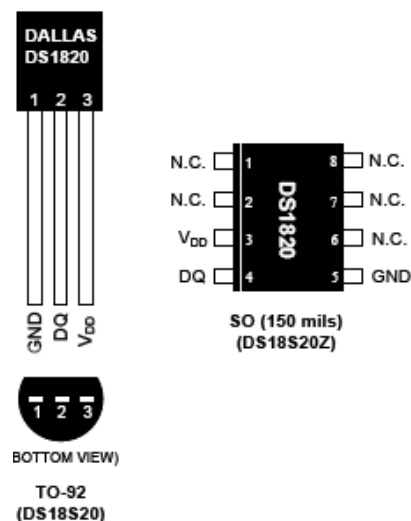
START	Aby mohl master zahájit komunikaci, musí vygenerovat START
STOP	Vygenerováním STOP master ukončuje komunikaci na sběrnici
Opakovaný START	Je využíván u některých slave zařízení, kde je například možno v průběhu komunikace vybrat zdroj dat nebo pokračovat ve specifické komunikaci, například po zaslání příkazu pro zařízení slave ihned provést čtení odezvy od tohoto zařízení. V takovém případě se negeneruje STOP ale provede se opakovaný START, což dá slave zařízení informaci o dalším požadavku bez toho aby byla ukončena komunikace podmínkou STOP. Současně je zabráněno uvolnění sběrnice, tudíž jiné master zařízení nemůže přerušit požadovanou komunikaci. Jedním takovým zařízením, vyžadujícím pro svoji komunikaci opakovaný START je hardwarový most DS2482-100.
DATA	Komunikační data synchronizuje master zařízení časováním na lince SCL.
Potvrzení ACK	Slave zařízení potvrzuje každý přijatý bajt od master zařízení.
Nepotvrzení ACK	Pokud slave zařízení nemůže přijmout nebo vyslat data, nepotvrdí příjem signálem ACK. V tomto případě nemusí potvrdit ani svoji adresu. Slave zařízení, které je připraveno komunikovat, potvrdí nejméně svoji adresu. Nicméně později může odmítnout poslaná data, například kvůli neplatnému příkazu, a nepotvrdí žádný z následujících bajtů. V tomto případě může master nejdříve generovat opakovaný START nebo STOP a následný START pro poslání nových dat.

*Tabulka 6-1: Komunikační události I<sup>2</sup>C sběrnice.*

Detailnější popis sběrnice I<sup>2</sup>C lze nalézt v literatuře [3] a specifikaci [4].

### 6.3. Teplotní čidlo DS18S20

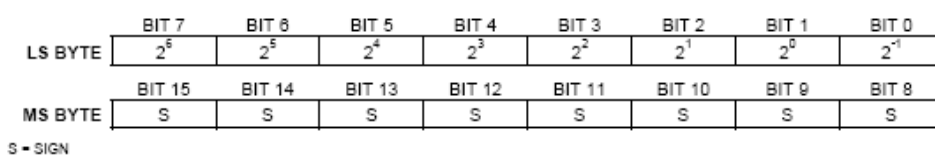
DS18S20 je teplotní čidlo (digitální teploměr), poskytující 9-bitové měření teploty ve stupních Celsia. Připojuje se k sběrnici 1-Wire, která je schopna datové komunikace pouze po jednom vodiči (datová linka, časování, napájení Parasite Power), druhý vodič je zem. Každé čidlo má jedinečný 64-bitový sériový kód uložený v interní ROM paměti, což umožňuje více čidlům pracovat na společné 1-Wire sběrnici. Čidlo nevyžaduje žádné externí komponenty a může být napájeno přímo z datové linky (tento návrh využívá oddělené napájení třetím vodičem). V napájecím režimu čidla od 3.0V to 5.5V je schopno měřit teploty od  $-55^{\circ}\text{C}$  do  $+125^{\circ}\text{C}$ , ale s přesností  $\pm 0.5^{\circ}\text{C}$  jen v rozsahu  $-10^{\circ}\text{C}$  až  $+85^{\circ}\text{C}$ . Maximální doba měření teploty čidlem je 750ms. Zajímavou funkcí čidla je možnost definování minimální a maximální teploty, při jejímž překročení si tuto událost čidlo zapamatuje a pokud je čidlu poslán funkční příkaz Alarm Search, potvrdí že u něj tento stav nastal (v tomto návrhu funkce nevyužita). Zapojení vývodů DS18S20 jak v pouzdře TO-92 tak i v pouzdře SO je na obrázku 6.4.



Obr. 6.4: Zapojení vývodů DS18S20.

### 6.3.1 Měření teploty

Hlavní funkcí DS18S20 je měření teploty a její přímý převod do digitální podoby. Výstupní změřená teplota představuje teplotu v krocích  $0.5^{\circ}\text{C}$  a má rozlišení 9 bitů. Po přivedení napájení je DS18S20 ve stavu nízkého příkonu. Pro zahájení A-D převodu teploty na digitální hodnotu, master zařízení musí poslat funkční příkaz Convert T[44h]. Ihned po převodu jsou výsledná teplotní data uložena v interní paměti ve 2 bajtovém registru a čidlo přechází opět do stavu nízkého příkonu. Teplotní data jsou cejchována ve stupních Celsia a jsou uložena jako dvojkový doplněk se znaménkem. Rozložení jednotlivých bitů 2 bajtového registru teploty je na obrázku 6.5.



Obr. 6.5: Registr teploty DS18S20.

Nižší bajt v bitech 7 až 1 obsahuje informaci o teplotě. Bit 0 obsahuje informaci o půl stupni Celsia, logická jednička v tomto bitu tedy značí  $0.5^{\circ}\text{C}$ , logická nula  $0.0^{\circ}\text{C}$ . Vyšší bajt obsahuje informaci o znaménku teplotních dat. Nastavení všech bitů do logické jedničky značí naměření záporné teploty, nastavení všech bitů do logické nuly značí naměření kladné nebo nulové teploty. Příklad obsahu registru DS18S20 pro různé teploty je na obrázku 6.6.

TEMPERATURE ( $^{\circ}\text{C}$ )	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+85.0*	0000 0000 1010 1010	00AAh
+25.0	0000 0000 0011 0010	0032h
+0.5	0000 0000 0000 0001	0001h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1111	FFFFh
-25.0	1111 1111 1100 1110	FFCEh
-55.0	1111 1111 1001 0010	FF92h

\*The power-on reset value of the temperature register is  $+85^{\circ}\text{C}$ .

Obr. 6.6: Příklad naměřených teplot.

### 6.3.2 64-bitový sériový kód

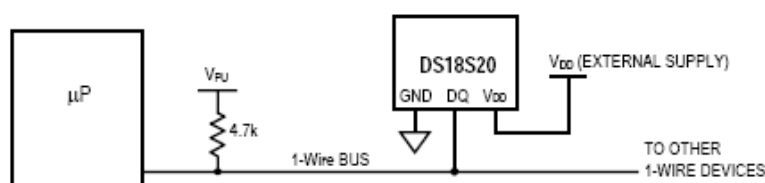
Každé čidlo DS18S20 obsahuje 64-bitový sériový kód, jehož jedinečnost je výrobcem garantována. Nejvyšších 8-bitů obsahuje kód skupiny (pro DS18S20 je to 10h), následujících 48-bitů je jedinečné sériové číslo a posledních 8-bitů je CRC součet prvních 56 bitů sériového kódu.

### 6.3.3 Interní paměť

Kromě již zmiňovaných 64-bitů ROM, obsahuje čidlo také 2 bajty SRAM pro uložení naměřené teploty a 2 bajty EEPROM pro uložení minimální a maximální teploty pro funkční příkaz Alarm Search. Dále jsou zde specifické registry pro doplňkové funkce a CRC. Paměť má celkem 9 bajtů.

### 6.3.4 Blokové schéma zapojení

DS18S20 používá exkluzivní 1-Wire sběrniceový protokol, který implementuje komunikaci po sběrnici použitím jediného řídicího signálu. Řídicí linka musí být napojena na pullup rezistor, protože všechny zařízení jsou na sběrnici připojena přes port s otevřeným kolektorem nebo přes 3-stavový port. Schéma připojení je uvedeno na obrázku 6.7.



Obr. 6.7: Připojení DS18S20 ke sběrnici.

### 6.3.5 Transakční protokol DS18S20

Protokol pro zpřístupnění DS18S20 na datové sběrnici vyžaduje provedení kroků Inicializace, posláni příkazu ROM a posláni příkazu Funkce. Tato sekvence musí být zachována při každém přístupu DS18S20, protože obvod nebude odpovídat, pokud bude některý krok vynechán nebo budou kroky přehozeny.

### 6.3.6 Inicializační sekvence

Inicializační sekvence zahrnuje reset pulse vyslaný master zařízením a následný presence pulse vyslaný slave zařízením DS18S20. Obdržením presence pulse je master upozorněn, že jsou na sběrnici dostupná slave zařízení a připravena odpovídat.

### 6.3.7 příkaz ROM

Po inicializační sekvenci může master zařízení vyslat příkaz ROM. Tento příkaz operuje nad jedinečným 64-bit ROM kódem každého slave zařízení na sběrnici a umožňuje tak master zařízení komunikovat s konkrétním slave zařízením. DS18S20 reaguje celkem na pět 8-mi bitových příkazů ROM, v návrhu je ale využit pouze jeden 55h.

MATCH ROM [55h] - posláním příkazu 55h s 64-bit ROM kódem umožňuje master zařízení adresovat na sběrnici konkrétní DS18S20 s odpovídajícím kódem. Pouze toto adresované slave zařízení odpoví na následující funkční příkaz vyslaný master zařízením, všechna ostatní slave zařízení na sběrnici budou čekat až na reset pulse.

### 6.3.8 příkaz Funkce

DS18S20 reaguje celkem na šest 8-mi bitových funkčních příkazů, v návrhu jsou ale využity pouze dva 44h a BEh. Tyto příkazy posílá master zařízení po příkazu ROM. Funkční příkazy umožňují master zařízení zapisovat a číst z paměti DS18S20 či zahájit měření teploty.

CONVERT T [44h] - funkční příkaz 44h zahájí měření teploty. Výsledná teplota je uložena v 2 bytovém registru v paměti DS18S20. Po tomto příkazu se čidlo uvede do stavu nízké spotřeby.

READ SCRATCHPAD [BEh] - funkční příkaz BEh umožňuje master zařízení přečíst obsah paměti DS18S20. Přenos dat začíná nejméně významným bitem bajtu 0 a pokračuje až po bajt 9. Master zařízení může poslat reset pulse a přerušit tak posílání dat v kterémkoliv okamžiku, pokud potřebuje přenést jen určitý počet bajtů. V návrhu přenášíme pouze první dva bajty obsahující změřenou teplotu.

### 6.3.9 Příklad komunikace

Příklad použité komunikace při čtení teploty z čidla DS18S20 je uveden v tabulce 6-2.

MASTER	DATA	POZNÁMKA
Tx	Reset	master zařízení pošle reste pulse
Rx	Presence	DS18S20 odpoví s presence pulse
Tx	55h	master zařízení pošle ROM příkaz match rom 55h
Tx	64-bit ROM kód	master zařízení adresuje DS18S20 pomocí 64-bit ROM kódu
Tx	44h	master zařízení pošle funkční příkaz 44h pro změření teploty
-	-	master čeká na dokončení měření 750ms
Tx	Reset	master zařízení pošle reste pulse
Rx	Presence	DS18S20 odpoví s presence pulse
Tx	55h	master zařízení pošle ROM příkaz match rom 55h
Tx	64-bit ROM kód	master zařízení adresuje DS18S20 pomocí 64-bit ROM kódu
Tx	BEh	master zařízení pošle funkční příkaz BEh pro čtení paměti
Rx	2 datové bajty	DS18S20 začne posílat 9 bajtů obsahu paměti
Tx	Reset	master zařízení přeruší přenos, potřebuje jen 2 první bajty
Tx = transmit (poslání), Rx = recieve (příjem)		

*Tabulka 6-2: Příklad čtení teploty.*

Detailnější informace včetně elektrické specifikace obvodu DS18S20 a jeho podporovaných funkcí lze nalézt v datasheetu v literatuře [5].

#### 6.4. Hardwarový most DS2482-100

Protože časování 1-Wire sběrnice má přísné podmínky, firma Maxim vyrábí různé komunikační obvody, které zajišťují rozhraní mezi 1-Wire sběrnici a jejím protokolem. Jedním takovým obvodem je DS2482-100. Tento obvod zajišťuje konverzi protokolu I<sup>2</sup>C na protokol 1-Wire. Obvod se připojuje přímo k hostiteli vybavenému sběrnici I<sup>2</sup>C, v našem případě modulu XT75 a provádí obousměrný převod mezi I<sup>2</sup>C sběrnici a každým slave zařízením připojeným na sběrnici 1-Wire. Každému slave zařízení na 1-wire sběrnici, se tento obvod jeví jako 1-Wire master zařízení. Obvod DS2482-100 je schopen zpracovat jak standardní I<sup>2</sup>C rychlost 100kHz, tak vyšší rychlost 400kHz a umí pracovat se standardní i overdrive rychlostí sběrnice 1-Wire.

V praxi použití DS2482-100 znamená, že pomocí I<sup>2</sup>C komunikace lze na sběrnici 1-Wire zapisovat data a řídicí příkazy a zpětně lze data získávat jejich čtením. Tím programátorovi odpadá těžký úkol řešení přesných časových souvislostí sběrnice 1-Wire, řešení kolizí a ostatních komunikačních událostí, ale také řešení teplotních závislostí, které ovlivňují časování na sběrnici 1-Wire. Další výhodou je zejména eliminace speciálního ladicího hardware, jako osciloskopu nebo logického analyzátoru, který by byl nutný k ověření správné komunikace sběrnice 1-Wire v různých režimech koncové aplikace, protože komunikaci po sběrnici ovlivňuje také fyzické chování sběrnice v reálném prostředí a její topologické uspořádání.

Na obrázku 6.8 je zjednodušené schéma, ve kterém DS2482-100 pracuje jako most mezi I<sup>2</sup>C portem modulu XT75 a 1-Wire sběrnici, ke které jsou připojeny slave zařízení.

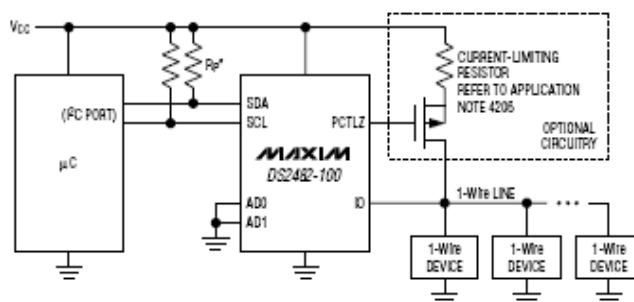


Obr. 6.8: Připojení DS2482-100 ke sběrnicím.



#### 6.4.1 Shrnutí vlastností obvodu

DS2482-100 podporuje 100kHz a 400kHz I<sup>2</sup>C rychlost komunikace, chová se jako 1-Wire master zařízení pro standardní i overdrive komunikační rychlost, je vybaven výstupem pro externí pullup MOSFET tranzistor a pomocí dvou adresních vstupů lze přidělit adresu I<sup>2</sup>C. Rozsah pracovního napájení je 2.9V až 5.5V, a pracovní teplota v rozmezí -40°C to +85°C. Typický příklad zapojení je na obrázku 6.9.



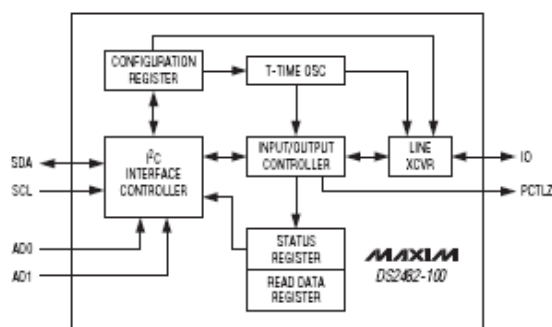
Obr. 6.9: Příklad zapojení DS2482-100.

#### 6.4.2 Funkce DS2482-100

DS2482-100 je master zařízení, které generuje časování pro sběrnici 1-Wire. Po tom co je sběrnici předán příkaz nebo data, vstupně/výstupní řídicí obvod DS2482-100 provede přísné časování na 1-Wire sběrnici pro reset, presence pulse, čtení bajtu, zápis bajtu, čtení a zápis jednotlivého bitu a tříbitové slabiky pro vyhledávací ROM příkaz, vše nezávisle a bez spolupráce hostujícího procesoru, v našem případě XT75. Host obdrží až výsledek požadované operace prostřednictvím Status registru nebo data přes Read Data registr. DS2482-100 komunikuje s hostem přes rozhraní sběrnice I<sup>2</sup>C ve standardním nebo rychlém režimu. Logická hodnota na adresních vývodech obvodu určuje I<sup>2</sup>C slave adresu DS2482-100. Tím lze připojit až 4 obvody a vytvořit tak 4 samostatné větve 1-Wire sběrnice bez použití hub zařízení.

#### 6.4.3 Blokové schéma DS2482-100

DS2482-100 obsahuje tři pracovní registry. Registry se adresují pomocí ukazatele při sekvenčním přístupu čtení. Pozice ukazatele je definována instrukcí, kterou DS2482-100 naposledy provedl. Blokové vnitřní zapojení obvodu včetně registrů je shrnuto na obrázku 6.10.



Obr. 6.10: Registry obvodu DS2482-100.

#### 6.4.4 Konfigurační registr

DS2482-100 podporuje tři vlastnosti 1-Wire sběrnice, které se povolují nebo zakazují přes konfigurační registr. Tyto vlastnosti jsou aktivní pullup (APU), silný pullup (SPU) a rychlost 1-Wire (1WS). Přivedením napájení na obvod nebo příkazem reset je konfigurační registr vynulován a musí být dle požadavku nastaven. Při zápisu do konfiguračního registru jsou data akceptována pouze pokud horní 4 bity jsou jednotkovým doplňkem spodních, ale při zpětném čtení konfiguračního registru jsou horní 4 bity vždy nulové.

#### 6.4.5 Status registr

Status registr je určen pouze pro čtení a obsahuje důležité informace pro host zařízení. Informace jsou v podobě jednotlivých bitů – příznaků. Všechny 1-Wire komunikační příkazy a příkaz reset zařízení, nastaví ukazatel čtení na status registr, aby měl host k těmto informacím okamžitý přístup s minimální režii jeho práce. Seznam bitů status registru je uveden v tabulce 6-3.

#### 6.4.6 Data Registr

Do registru dat jsou ukládána data pro host zařízení, jako výsledek funkčních příkazů. Ukazatel čtení je automaticky nastaven na registr dat po každém funkčním příkazu, aby mohl host zařízení efektivně a rychle číst požadované informace.

1-Wire Busy (1WB)	Status obsazené linky sběrnice 1-Wire. Příznak 1WB = 1 indikuje obsazenou linku nebo právě probíhající příkaz na sběrnici 1-Wire, který ještě nebyl dokončen.
Presence-Pulse Detect (PPD)	Status je obnovován po každém vyslání příkazu reset pro 1-Wire sběrnici. Pokud je detekován presence pulse kterýmkoliv ze slave zařízení, je příznak nastaven na 1, PPD = 1. Do výchozí hodnoty nula se vrátí, pokud nebyl presence pulse detekován nebo dalším zavoláním příkazu 1-Wire reset.
Short Detected (SD)	Status je obnovován po každém vyslání příkazu reset pro 1-Wire sběrnici. Pokud je detekována logická nula v určitém úseku průběhu detekce presence pulse, je příznak nastaven na 1, SD = 1. Do výchozí hodnoty nula se vrátí dalším zavoláním příkazu 1-Wire reset.
Logic Level (LL)	Tento status vrací logický stav linky 1-Wire bez poslání jakéhokoliv příkazu. Tento příznak je vzorkován pokaždé, kdy je čten status registr, DS2482-100 je tedy v režimu čtení a tedy ukazatel čtení je vždy nastaven na status registr.
Device Reset (RST)	Status je nastaven na 1 vždy když byl proveden reset, tedy buď po zapnutí napájení nebo vykonáním příkazu reset zařízení. Bit je nastaven na nulu automaticky po provedení příkazu zápisu do konfiguračního registru a tedy nastavení pracovního režimu DS2482-100.
Single Bit Result (SBR)	Tento příznak vrací logický stav na 1-Wire sběrnici po vykonání 1-bitového nebo 3-bitového příkazu pro vyhledávací algoritmus připojených slave zařízení.
Triplet Second Bit (TSB)	Tento příznak vrací logický stav druhého bitu 3-bitového 1-Wire příkazu a ostatní funkce nemají na tento bit vliv.
Branch Direction Taken (DIR)	Kdykoliv je spuštěn 3-bitový 1-Wire příkaz, tento příznak dává na vědomí hostujícímu zařízení směr vyhledávání, který byl zvolen třetím bitem 3-bitového příkazu a ostatní funkce nemají na tento bit vliv.

*Tabulka 6-3: Význam bitů status registru.*

#### 6.4.7 Funkční příkazy

DS2482-100 rozpoznává osm funkčních příkazů rozdělených do čtyř kategorií. Jde o příkazy na ovládání obvodu, komunikaci I<sup>2</sup>C, nastavení sběrnice 1-Wire a komunikaci na 1-Wire sběrnici. Host zařízení, v našem případě modul XT75, pošle tyto funkční příkazy jako jedno nebo dvoubajtový řetězec přes I<sup>2</sup>C rozhraní. Přehled příkazů je v tabulce 6-4.

PŘÍKAZ	DATA	FUNKCE
F0h	-	reset DS2482-100
E1h	ukazatel paměti	E1h F0h nastaví ukazatel čtení na Status registr E1h E1h nastaví ukazatel čtení na Data registr E1h C3h nastaví ukazatel čtení na konfigurační registr
D2h	bajt	zápis do konfiguračního registru
B4h		reset 1-Wire sběrnice
87h	bajt	zápis jednoho bitu na 1-Wire sběrnici, hodnota je udána v sedmém bitu bajtu
A5h	bajt	zápis bajtu na 1-Wire sběrnici
96h		čtení bajtu z 1-Wire sběrnice
78h	bajt	pro provedení 1-Wire ROM vyhledání na sběrnici 1-Wire

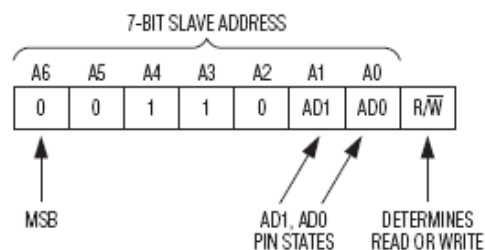
Tabulka 6-4: Funkční příkazy.

#### 6.4.8 I<sup>2</sup>C rozhraní

I<sup>2</sup>C sběrnice používá pro komunikaci datovou linku (SDA) plus hodinový signál (SCL). Obě linky jsou obousměrné a napojené přes pullup rezistor na kladné napájení. Pokud na sběrnici I<sup>2</sup>C neprobíhá žádná komunikace, obě linky jsou ve stavu vysoké úrovně. Na každé slave zařízení se odkazuje master zařízení pomocí jedinečné adresy na sběrnici I<sup>2</sup>C. V našem případě je modul XT75 master zařízením vůči slave zařízení DS2482-100.

#### 6.4.9 Adresa DS2482-100

Logická hodnota na vstupech AD0 a AD1 obvodu DS2482-100 určuje jeho adresu na sběrnici I<sup>2</sup>C, jak je znázorněno na obrázku 6.11. Tím je umožněno adresovat obvod až 4-mi různými adresami. Adresa potom spolu s příznakem směru dat (R/W) tvoří řídicí bajt vysílaný master zařízením pro nalezení adresáta na sběrnici I<sup>2</sup>C, v našem případě obvodu DS2482-100.



*Obr. 6.11: Adresace obvodu DS2482-100.*

Pokud je bit R/W v adrese nastaven na 0, všechna následná data od master zařízení (modulu XT75) budou určena pro slave zařízení (DS2482-100 režim zápisu). Pokud je bit R/W v adrese nastaven na 1, bude modul XT75 očekávat data od slave zařízení (DS2482-100 režim čtení).

#### 6.4.10 Zápis do DS2482-100

Zápis provede master zařízení XT75 posláním správné adresy s příznakem pro zápis na port I<sup>2</sup>C sběrnice a následně pošle bajt pro příkaz a případně navíc další bajt dat, pokud jej komunikační protokol pro slave zařízení vyžaduje. DS2482-100 odpoví potvrzením platného příkazu a potvrzením případného dalšího bajtu dat. Všechny nadbytečné bajty nerozpoznávané slave zařízením potvrzeny nejsou.

#### 6.4.11 Čtení z DS2482-100

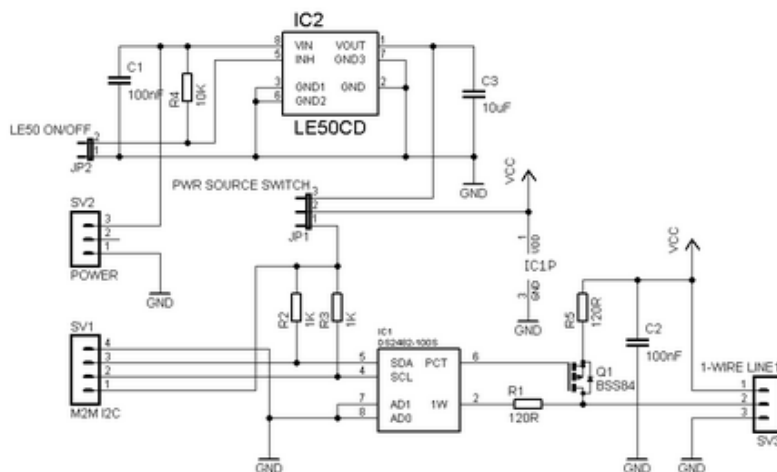
Čtení provede master zařízení XT75 posláním správné adresy s příznakem pro čtení na port I<sup>2</sup>C sběrnice. To, ze kterého registru bude DS2482-100 číst a následně zasílat data zpět master zařízení, ovlivňuje ukazatel čtení. Master může číst data opakovaně bez přeadresování DS2482-100, což je vhodné například pro sledování změny nějakého příznaku ve status registru. Pokud je třeba číst z jiného registru, master musí poslat příkaz pro nastavení ukazatele čtení a potom opět příkaz pro čtení DS2482-100.

Detailnější informace včetně elektrické specifikace obvodu DS2482-100 a jeho podporovaných funkcí lze nalézt v datasheetu v literatuře [6]. Popis komunikace s obvodem DS2482-100 lze nalézt v literatuře [10].

## 7. Hardwarový návrh

### 7.1. Schéma zapojení modulu DS2480-100

Vzhledem k tomu, že jde o první návrh s obvodem DS2482-100, bylo počítáno v konstrukci s univerzálním zapojením. Navržené zapojení umožňuje napájení 1-Wire sběrnice z modulu XT75 napájecím napětím 3.3V nebo lze použít externí napájení napětím 5-12V přivedeným na svorkovnici SV2. Zvolený způsob napájení řídí propojka JP1. Propojkou JP2 lze stabilizátor napětí LE50 úplně vyřadit z činnosti. Dále modul obsahuje dva komunikační konektory, jeden SV1 pro připojení ke sběrnici I<sup>2</sup>C modulu XT75 a druhý SV3 pro připojení sběrnice 1-Wire s napojenými čidly. Schéma zapojení je na obrázku 7.1.



Obr. 7.1: Schéma zapojení modulu DS2482-100.

Napěťový stabilizátor IC2 je v doporučeném zapojení. Obvod DS2482-100 má vstupy I<sup>2</sup>C sběrnice opatřeny pullup rezistory R2 a R3. Výstup na 1-Wire sběrnici je přes proud limitující rezistor R1. V zapojení je využit výstup PCTLZ strong pullup, napojený na externí tranzistor Q1. Tranzistor je aktivní, když je výstup PCTLZ v logické 0. Oproti internímu pullup rezistoru DS2480-100, obvod s tranzistorem Q1 poskytuje nižší úbytek napětí a poskytne vyšší proud do sběrnice 1-Wire. Prodlužuje se tak spolehlivost sběrnice při použití prvků, které pro svoji činnost vyžadují vyšší proud, například EEPROM 1-Wire paměti a zvětšuje se dostupná délka sběrnice 1-Wire. Kondenzátor C2 filtruje případné rušení v napájecí větvi 1-Wire sběrnice.

Po uvedení tohoto modulu do provozu a následných zkouškách bylo zjištěno, že velmi dobře a spolehlivě pracuje při napájení 3.3V přímo z modulu XT75. V testovacím nasazení tedy není využito napájení přes stabilizátor LE50 a tento obvod může být vypuštěn.

## 7.2. Stanovení pullup rezistoru I<sup>2</sup>C

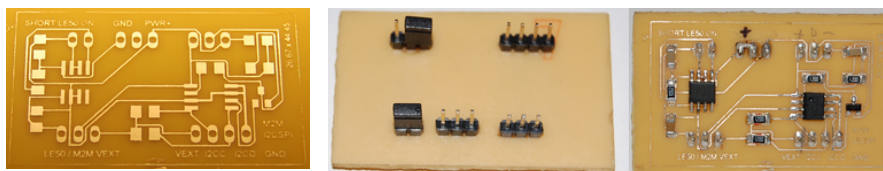
Pro správnou činnost sběrnice I<sup>2</sup>C je potřeba stanovit pullup rezistory pro linky SDA a SCL, které zajišťují na sběrnici napětovou úroveň logické jedničky, protože připojená zařízení spínají sběrnici buď výstupem open-drain nebo open-collector. Podle I<sup>2</sup>C specifikace je definováno, že výstupní napětí v nízké úrovni slave zařízení (sepnutá linka SDA nebo SCL) musí být 0.4V při minimálním protékajícím proudu 3mA, což obvod DS2482-100 podle elektrické specifikace splňuje. Daná podmínka I<sup>2</sup>C specifikace potom stanovuje minimální hodnotu pullup rezistoru  $R_{p(min)} = (V_{cc} - 0.4V)/3mA$ . Protože DS2482-100 je napájen napětím 3V z modulu XT75, minimální pullup rezistor vychází na 866 Ω. V zapojení byla pro rezistory R2 a R3 vybrána nejbližší hodnota z řady o velikosti 1 kΩ.

## 7.3. Návrh a výroba PCB

Pro návrh plošného spoje (PCB) byl použit freeware návrhový software firmy CadSoft ve verzi Light. Tento software umožňuje nakreslit návrh zapojení obvodu a z něj následně ručně nebo pomocí autorouteru vytvořit plošný spoj. Verze Light je limitována maximálním rozměrem desky 100 x 80 mm, dále lze navrhovat pouze dvouvrstvé PCB a editor schémat může mít pouze jeden návrhový list.

Samotná výroba PCB byla provedena v domácích podmínkách pozitivní fotocestou, což má výhodu v nízké ceně výroby, kvalitě fotocesty a je zde možnost rychle a opakovaně vyrobít více kusů v případě nutnosti změny zapojení obvodu. Podmínkou je mít odpovídající vybavení a omezení se na výrobu jednostranného plošného spoje bez prokovu. Celý postup výroby zabere přibližně 30-40minut, což je pro domácí podmínky velmi dobrý čas.

Na obrázku 7.2 je vyrobený plošný spoj ošetřený chemickou stříbricí lázní a následně osazený elektronickými součástkami.



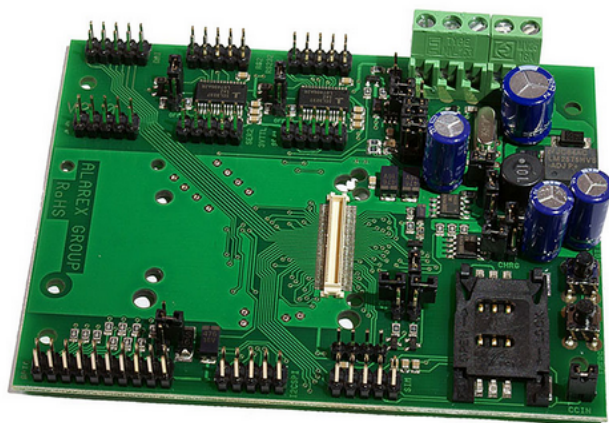
Obr. 7.2: Vyrobený a osazený PCB.

#### 7.4. Popis vývojového kitu M2M Board

Samotný modul XT75 má vyvedené veškeré sběrnice na SMD 80pin 3mm MOLEX, ke kterému je technicky nemožné připájet dráty pro vyvedení čehokoliv. Každý modul tedy vyžaduje použití základní desky, ke které se tímto konektorem připojí. Takovou základní desku lze navrhnout na míru pro potřeby koncového zařízení. Protože je návrh zpracováván ve směru vývojovém, nikoliv pro koncové nasazení, jako základní desku jsme použili M2M Universal Motherboard ve verzi 2.0 od firmy Alarex Group, s.r.o. Deska byla výrobcem navržena jako vývojový prostředek nebo základ GSM aplikací pro moduly Cinterion TC65, XT65, AC75, XT75 a další a její vyobrazení je na obrázku 7.3.

Tato základní deska má vyvedeny všechny rozhraní použitého modulu z konektoru Molex na konektory umístěné po obvodu desky. Deska dále obsahuje zdroj pro napájení modulu a obvody pro dobíjení připojené baterie. Součástí desky je držák SIM karty.

Základní deska M2M Universal Motherboard je výrobcem primárně určena pro vývoj aplikací, ladění, testování modulů, přehrávání firmware, ale také jako základ malosériových zařízení, kde se nevyplatí návrh a výroba specializovaných desek plošných spojů. Rozměry desky jsou 89x107x21mm a požadované napájecí napětí desky je v rozsahu 7-60 V.



*Obr. 7.3: M2M Universal Motherboard 2.0.*



Souhrnně je M2M Universal Motherboard vybavena takto:

- konfigurovatelný watchdog, čas watchdogu lze nastavovat v rozsahu 10-150s nebo 1-15 minut, watchdog lze také zcela vyřadit z činnosti. Firmware watchdogu lze měnit podle přání zákazníka
- pro oživování watchdogu lze použít GPIO 9 nebo 10 modulu
- dva kanály rozhraní RS232, jejichž napětovou úroveň lze u každého kanálu samostatně nastavit na 3V TTL nebo standardní úroveň RS232. Sériové linky v režimu RS232 nyní fungují i při napájení z baterie
- restart GSM modulu je proveden vypnutím napájení na dobu 1 minuty
- externí konektor pro stavovou LED GSM modulu – SYNC Led

#### 7.5. Realizace I<sup>2</sup>C na modulu XT75 přes linky GPIO

Modul XT75 je vybaven vestavěnou I<sup>2</sup>C sběrnici, která je použitelná jako master zařízení a která je ovládaná pomocí rozšířených AT příkazů modulu XT75. S touto vestavěnou sběrnici bylo od počátku návrhu počítáno pro připojení modulu DS2482-100. V průběhu vývoje aplikace modulu XT75 v jazyku Java, však došlo ke komplikacím. Bylo možno navázat komunikaci s modulem DS2482-100, ale nebyla možná žádná další komunikace a vyslání jednotlivých příkazů pro sběrnici 1-Wire. Po hodinách ladění bylo nakonec zjištěno, že modul XT75 vysílá požadavek na sběrnici I<sup>2</sup>C až po uzavření celého datového požadavku mezi šipky „<“ a „>“ (I<sup>2</sup>C příkaz pro START a STOP). A zde nastal problém, protože modul DS2482-100 vyžaduje ve specifikaci I<sup>2</sup>C schopnost poslat příkaz opakovaný start (REPEATED START).

Při prvotním návrhu a výběru vhodného hardware, byla podrobně studována specifikace modulu XT75, kde je zmíněna specifikace zástupných znaků „<“ a „>“ pro start a stop. Ovšem nebyla zde uvedena poznámka, že modul nepodporuje specifikaci restart například dvojitým použitím znaku „<“. Proto bylo původně zamýšleno, že datová zpráva pro I<sup>2</sup>C port může vypadat s restartem například jako „< 01 02 A4 < B4 00 10 >“. Ve zprávě druhá levá šipka by mohla značit restart. Modul XT75 ale tuto zprávu vyhodnotí jako data mezi šipkami, tedy jako „01 02 A4 < B4 00 10“, což samozřejmě pro modul DS2482-100 je neplatný požadavek. Rozdělit zprávu na dvě části jako „< 01 02 A4 >“ a „< B4 00 10 >“ nelze, neboť po podmínce stop je nutná nová adresace slave zařízení s jeho adresou, a tudíž slave zařízení již nepočítají s přísunem nových dat a data odmítnou jejich nepotvrzením.

Přes různé možnosti řešení tohoto nedostatku modulu XT75 byl nakonec zvolen pokus o realizaci napsání vlastní softwarové I<sup>2</sup>C rutiny, která bude emulovat chování I<sup>2</sup>C sběrnice a jejího protokolu přes dostupné dva GPIO piny modulu XT75. Tento krok se nakonec ukázal jako úspěšný a funkční, nicméně s rychlostním omezením díky ovládání GPIO pinů.

## 8. Softwarový návrh

### 8.1. Popis vývojového prostředí

Vzhledem k tomu, že modul XT75 se programuje pomocí programovacího jazyka Java, konkrétně verze J2ME s profilem IMP-NG, byl pro tvorbu ovládacího programu modulu XT75 zvolen software NetBeans IDE od společnosti SUN ve verzi 9.7.1

Dále je nutné nainstalovat podpůrné CD modulu XT75, kde kromě všech hardwarových specifikací a pdf manuálů, jsou taktéž ovladače modemu pro připojení modulu XT75 přes RS232 k vývojovému PC a taktéž je zde dostupný profil IMP-NG pro integraci do prostředí Netbeans IDE. Dále je zde dostupný ovladač, který v operačním systému vytvoří novou ikonu zařízení Module.

Modul XT75 je přes základní desku M2M Universal MotherBoard 2.0 připojen k portu RS232 vývojového PC a k základní desce je připojeno externí napájecí napětí. Systém, v našem případě Windows XP, detekuje nový modem ve správci zařízení. Od této chvíle se stane modul dostupný v operačním systému pod ikonou Module a je možné zobrazit obsah jeho interní Flash paměti. V tabulce 8-1 je zobrazen obsah paměti Flash modulu XT75. Jsou zde vidět programové soubory M2M\_1wire.jad a M2M\_1wire.jar, dále konfigurační soubor config.txt a datové soubory s aktuálně naměřenými teplotami ve formátu csv a xml. s nahráním programem.

config.txt	124 bytes
M2M_1wire.jad	224 bytes
M2M_1wire.jar	13,8 KB
temperature.csv	1,38 KB
temperature.xml	245 bytes

*Tabulka 8-1: Obsah Flash paměti modulu XT75.*

Ikona zařízení Module se v operačním systému chová jako paměťové zařízení, a lze s ním pracovat jako s běžnými souborovými prostředky.

Po odladění kódu v NetBeans IDE je potřeba jej přenést do modulu XT75. To lze provést po kompilaci kódu ručním zkopírováním souborů M2M\_1wire.jad, M2M\_1wire.jar a config.txt do paměti Flash modulu přes ikonu zařízení Module nebo se o nahrání do Flash paměti postará NetBeans IDE a nahráný kód ihned spustí. Tyto vlastnosti jsou v prostředí NetBeans IDE konfigurovatelné. Ruční spuštění se provádí přes konzoli RS232 (například Hyperterminál) příkazem AT^SJRA=A:/M2M\_1wire.jar. Pro spuštění lze také použít soubor jad.

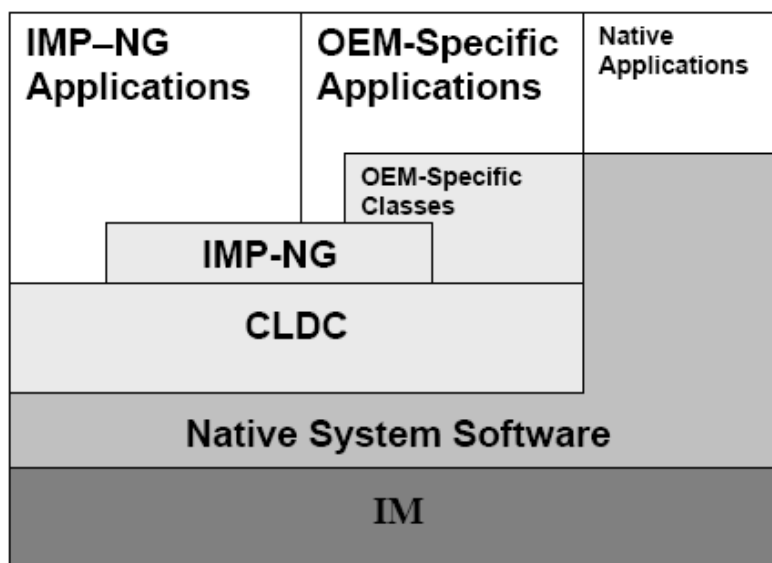
## 8.2. Java ME

Platforma Java Micro Edition (Java ME) poskytuje rozsáhlé a flexibilní prostředí pro aplikace běžící na mobilních nebo vestavěných zařízeních jako jsou mobilní telefony, PDA, TV sety nebo dokonce tiskárny. Java ME zahrnuje přizpůsobitelná uživatelská rozhraní, bezpečnost, síťové protokoly a možnost aplikace stahovat přes síť. Aplikace založené na Java ME jsou přenositelné mezi zařízeními, pokud mají stejné hardwarové schopnosti.

Každé zařízení Java ME musí implementovat profil. Nejběžnějším profilem, který je dnes implementován v mobilních telefonech je MIDP ve verzi 2.0. Profily jsou podmnožinou konfigurací, které jsou v současné době dvě, CDC a CLDC. Pro mobilní zařízení se používá Limitovaná verze CDC, tedy CLDC (Connected Limited Device Configuration). Konfigurace CLDC obsahuje podskupinu Java tříd a knihoven a je to nejmenší možná část, která je nutná pro spuštění JVM (Java Virtual Machine).

Celé zařízení tedy musí v konečné fázi obsahovat konfiguraci a profil. Profil MIDP 2.0 (Mobile Information Device Profile) obsahuje rozšířené uživatelské rozhraní, multimediální a 2D API pro hry, grafické GUI API, výbornou síťovou konektivitu, vzdálené stahování a instalování aplikací (OTA provisioning) a bezpečnostní model. Aplikace napsané v tomto profilu se nazývají Midlety.

Modul XT75 podporuje konfiguraci CLDC a profil IMP-NG (JSR-228). Tento profil vychází z profilu MIDP 2.0, neobsahuje ale grafické API (`javax.microedition.lcdui`), protože modul nemá zobrazovací displej a API pro hry.



*Obr. 8.1: Platforma J2ME.*

Na obrázku 8.1 je zobrazena J2ME platforma. Implementovaný modul IM zahrnuje hardware modulu XT75. Nad ním je postaven NSS což je operační systém a knihovny, které modul používá. V tomto operačním systému běží JVM a konfigurace podle specifikace CLDC. Až potud se jedná o základní systém, na kterém se teprve staví specifické vlastnosti zařízení jako profil IMP-NG a specifické třídy pro každé zařízení. Pro modul XT75 jsou specifickými třídami AT Command API pro ovládání hardwarového vybavení jako například I<sup>2</sup>C nebo GPIO a File I/O API pro přístup k paměti Flash modulu XT75.

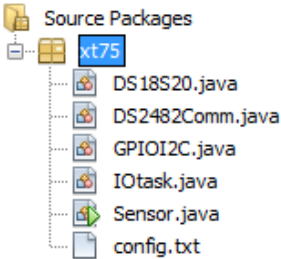
V této práci se využívají následující prostředky modulu XT75:

- File I/O API pro přístup k paměti Flash modulu XT75
- AT Command API pro přístup k nastavení RTC času modulu
- AT Command API pro ovládání vstupně/výstupních pinů GPIO
- HTTP spojení pro synchronizaci času
- HTTP spojení pro odeslání dat na server

### 8.3. Popis software modulu XT75

Zdrojový balík J2ME obsahuje konfigurační soubor a několik tříd, které se dají rozdělit na jednotlivé části podle tabulky 8-2.

config.txt	konfigurační soubor modulu XT75
Sensor.java	hlavní programová část
DS2482Comm.java DS18S20.java	ovládání externího hardware
GPIOI2C.java	ovládání hardware modulu XT75
IOTask.java	komunikační část



Tabulka 8-2: Skladba software modulu XT75.

#### 8.3.1 *config.txt*

Konfigurační soubor slouží modulu XT75 pro zadání hardwarových adres teplotních senzorů a pro nastavení výstupního formátu, ve kterém budou naměřené teploty odesílány na XML server ke zpracování. Příklad konfiguračního souboru je v tabulce 8-3

```
Sensor 1:150008018F317810:Inside  
Sensor 2:DB0008018F382610:Heater  
Sensor 3:5D0008018F572D10:Outside  
OutputFileFormat :xml
```

Tabulka 8-3: Výpis konfiguračního souboru.

Každý senzor má vymezen samostatný řádek ve formátu „sensor : adresa : popis“. Pomocí značky „OutputFileFormat“ se nastavuje výstupní formát dat jako XML nebo CSV. Konfigurační soubor je umístěn ve Flash paměťovém prostoru modulu XT75 spolu se spouštěcími soubory.

### 8.3.2 *Sensor.java*

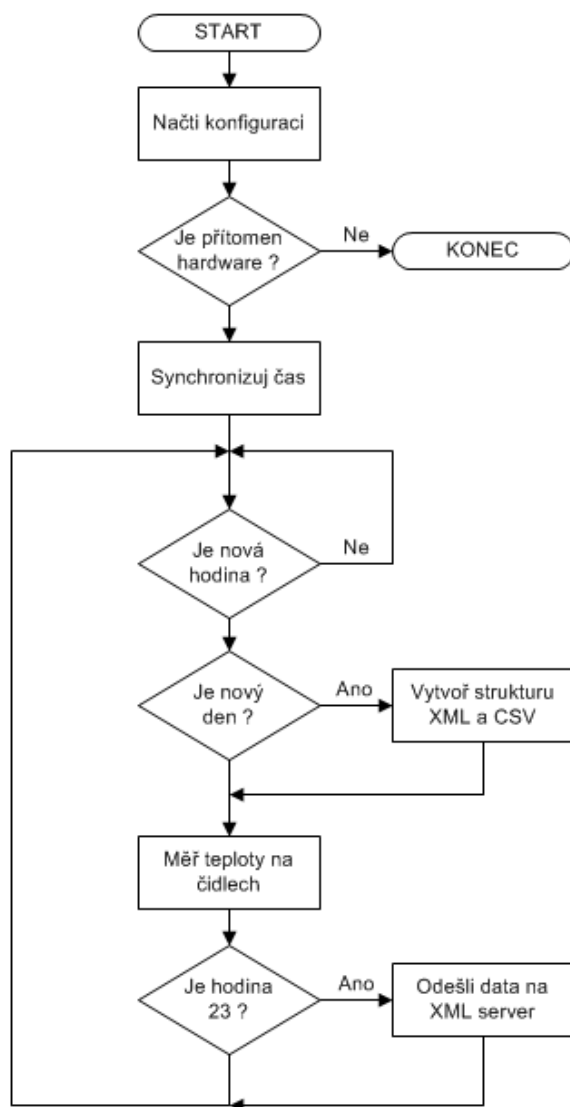
Třída *Sensor.java* je hlavní spouštěcí třídou ovládající chování modulu XT75. Její provádění kopíruje popis uvedený výše ve funkci zařízení, zde je popsána funkce z pohledu samotného programu.

Třída *Sensor.java* nejdříve přečte konfigurační soubor a uloží si informace o hardwarové adrese senzoru teploty, o jeho popisu a typu požadovaného výstupního souboru. Následně třída otestuje, zda-li je připojen příslušný hardware pro ovládání 1-Wire sběrnice, hardwarový most DS2482-100, viz. tabulka 8-4. Hardwarový most je volán po I<sup>2</sup>C sběrnici se standardně nastavenou hardwarovou adresou „0x30H“.

```
ow = new DS2482Comm();
if (ow.DS2482_Detect("30")) {
    System.out.println("Device DS2482 succesfully detected.");
    ow.DestroyComm();
} else {
    System.out.println("Device DS2482 was not detected !");
    ow.DestroyComm();
    destroyApp(false);
    notifyDestroyed();
}
```

*Tabulka 8-4: Test hardware.*

Pokud je hardwarový most úspěšně detekován, pokračuje třída *Sensor.java* nastavováním času zavoláním funkce *syncTime\_HTTP* třídy *Iotask.java*. Po synchronizaci času třída *Sensor.java* nejdříve čeká a každou minutu testuje, zda-li již nastala celá hodina. Tím je zajištěno měření teplot pouze v každou celou hodinu. Pokud nastala celá hodina a jde o první měření teploty ve dni, zapíše se datové hlavičky souboru CSV a vytvoří se struktura XML souboru. V průběhu vykonávání třídy *Sensor.java* se plní oba soubory daty naměřených teplot na třech teplotních senzorech. Po změření teplot třída *Sensor.java* zjišťuje hodinu, ve které právě měřila teploty. Pokud je hodina rovna 23, odešle modul XT75 všechny doposud naměřená data na XML server ke zpracování. Po odeslání dat je třída *Sensor.java* odkázána opět na svůj počátek, pracuje tedy v nekonečné smyčce. Protože splnění podmínky pro odeslání dat na XML server nastalo ve 23 hodin, nové měření teplot bude až v hodině 00:00, tedy v následujícím dni. Práce třídy *Sensor.java* je zachycena ve vývojovém diagramu na obrázku 8.2.



Obr. 8.2: Vývojový diagram třídy `Sensor.java`.

### 8.3.3 *DS2482Comm.java a GPIOI2C.java*

Třída DS2482Comm.java zahrnuje komunikační protokol s obvodem hardwarového mostu DS2482-100 přes protokol I<sup>2</sup>C. Ke své práci tato třída vyžaduje spolupráci s třídou GPIOI2C.java. Třída DS2482Comm.java je rozdělena na část, která řídí hardwarový most DS2482-100 a na část, která řídí komunikaci na sběrnici 1-Wire.

Řídící část hardwarového mostu obsluhuje metody DS2482\_Detect, DS2482\_Reset\_Device, DS2482\_Write\_Configuration a metodu DestroyComm.

Metoda DS2482\_Detect detekuje připojení obvodu hardwarového mostu DS2482-100 ke sběrnici I<sup>2</sup>C. Ve svém těle nejprve zavolá metodu DS2482\_Reset\_Device a následně metodu DS2482\_Write\_Configuration. Pokud tyto dvě metody skončí úspěšně, byl hardwarový most DS2482-100 úspěšně detekován a může začít probíhat další komunikace již na úrovni 1-Wire sběrnice.

Metody OWReset, OWWrite\_Byte a OWRead\_Byte jsou metody obsluhující základní komunikační protokol na sběrnici 1-Wire. Zavoláním metody OWReset vygeneruje hardwarový most DS2482-100 na sběrnici 1-Wire správné časování pro její reset a připraví ji tak na následující komunikaci. Metody OWWrite\_Byte a OWRead\_Byte potom zapisují bajt na 1-Wire sběrnici nebo z ní bajt čtou.

Metoda DestroyComm uzavírá AT command kanál otevřený při vytvoření instance třídy DS2482Comm. AT command kanál se používá pro ovládání vstupně/výstupních GPIO linek modulu XT75 ve třídě GPIOI2C.java. Toto uzavření je vyžadováno ve specifikaci modulu XT75, protože je možno otevřít pouze omezený počet AT command kanálů.

GPIOI2C.java je implementací softwarového protokolu I<sup>2</sup>C namapovaného na dvě hardwarové vstupně/výstupní GPIO linky modulu XT75, ke kterým je hardwarový most DS2482-100 fyzicky připojen. Tato třída je poměrně rozsáhlá, neboť implementuje jak funkce pro ovládání vstupně/výstupních GPIO linek pomocí AT příkazů, tak všechny příkazy a obsluhu protokolu sběrnice I<sup>2</sup>C. Pin GPIO1 modulu XT75 je přiřazen jako datová linka SDA a pin GPIO2 jako linka časování SCL.

Reálná rychlost I<sup>2</sup>C není softwarově nijak stanovena, je tedy dosahována maximální rychlost, jaké je schopen modul XT75 dosáhnout na svých GPIO pinech. Vzhledem k použití AT příkazů lze spíše mluvit o pomalosti. Reálnou komunikační rychlost po sběrnici I<sup>2</sup>C by bylo možno změřit v laboratoři například logickým analyzátozem. K softwarové implementaci I<sup>2</sup>C byly využity informace z PseudoKódu viz. literatura [7].



#### 8.3.4 DS18S20.java

DS18S20.java je třída obsluhující návaznost bajtových příkazů na sběrnici 1-Wire tak, aby ze stejnojmenného teplotního snímače DS18S20 mohla být získána naměřená teplota. Vstupem pro tuto třídu je hardwarová adresa teplotního snímače. Tato třída následně předává příkazy v podobě jednotlivých bajtů třídě DS2482Comm.java a ta za pomoci ovládání hardwarového mostu DS2482-100 generuje na sběrnici odpovídající 1-Wire signály.

Hlavním prvkem třídy DS18S20.java je metoda `get_TempC`, která se stará o změření teploty teplotním čidlem DS18S20 a vrácení této teploty ve stupních celsia.

Protože teplota je uložena v registrech teplotního čidla DS18S20 jako dvojkový doplněk, musí být převedena zpět na číselnou hodnotu. Příklad řešení je v tabulce 8-5.

```
int temp = 0;

int temp_byte_lo = Integer.parseInt(ow.OWRead_Byte(), 16);
int temp_byte_hi = Integer.parseInt(ow.OWRead_Byte(), 16);

if (temp_byte_hi == 0) {
    temp = temp_byte_lo >> 1;
} else {
    temp = ~((255-temp_byte_lo)>>1);
}
```

*Tabulka 8-5: Převod teploty z dvojkového doplněku.*

#### 8.3.5 IOTask.java

Třída IOTask.java zabezpečuje operace nad Flash pamětí modulu XT75, jako smazání souboru, zápis do souboru a čtení ze souboru. Tyto operace modul využívá k zaznamenávání teplot do souboru před tím, než jsou data odeslána na XML server. Další dvě metody slouží ke vnější komunikaci modulu XT75 a spojení přes GPRS. První metoda `write_HTTP` posílá data metodou POST na vzdálený server a metoda `syncTime_HTTP` synchronizuje čas serveru s časem uloženým v RTC modulu XT75.

#### 8.4. Získání atomového času J2ME/PHP

Protože modul XT75 pracuje s časovými údaji, potřebuje znát informace o aktuálním času. I když je modul vybaven obvodem RTC, není tento samostatně zálohován napájecí baterií. V praxi to znamená, že jakmile modul ztratí napájení, je nastavený čas nenávratně ztracen. Při dalším připojení napájecího napájení je RTC nastaven na výchozí hodnotu předdefinovanou výrobcem modulu na "02/01/01,00:00:00" ve formátu "yy/mm/dd,hh:mm:ss".

Z tohoto důvodu je v modulu XT75 naprogramována automatická synchronizace času ihned po připojení napájení. Čas je synchronizován přes internetové spojení GPRS na PHP skript, který získá čas ze serveru „time-a.nist.gov“ a vrátí modulu XT75 zpět čas metodou POST ve formátu "YmdHi", například „201003072308“. Veškeré zpracování tedy probíhá na straně PHP skriptu uvedeného v tabulce 8-6.

```
$server = "time-a.nist.gov";  
$fp = fsockopen($server, 37);  
$time = fread($fp, 4);  
fclose($fp);  
  
echo Date("YmdHi", hexdec(bin2hex($time)) - 2208988800);
```

*Tabulka 8-6: Získání atomového času.*

Číslo 2,208,988,800 vyjadřuje datum a čas od začátku Unixové epochy 00:00 1 Jan 1970 GMT. Atomový čas se vrátí jako číselná hodnota od roku nula, unixový čas je ale počítán až od začátku Unixové epochy. Proto je od vráceného atomového času odečten počátek Unixové epochy a veškeré operace v Unixovém času operačního systému, v našem případě i modulu XT75 budou následně ve správném čase.

J2ME program získává informace o čase z RTC obvodu modulu XT75 v době své inicializace a nikdy později. Snaha o úpravu času v programu J2ME zavoláním AT příkazu (AT+CCLK) je marná. Tímto příkazem se nastaví správný datum a čas obvodu RTC modulu XT75, nedojde ale k úpravě času v J2ME metody System.currentTimeMillis().

Proto metoda syncTime\_HTTP vrací offset času mezi atomovým časem získaným ze serveru a aktuálním časem v programu J2ME, metodě System.currentTimeMillis(). Následně kdykoliv je potřeba v J2ME znát aktuální čas, k metodě System.currentTimeMillis() se připočítá vykalkulovaný offset. Samozřejmě řešením by bylo získat čas AT příkazem (AT+CCLK) přímo z obvodu RTC modulu XT75 po tom, co je jednou správně nastaven. Tento způsob je ale časově náročný a programově zdlouhavý. V aplikacích, kde se čas používá velmi často, jako u tohoto návrhu, jde o realizaci přímo nevhodnou.

## 8.5. XML Server

Na straně centrálního datového centra, do kterého modul XT75 posílá data o změřených teplotách za uplynulý den, je realizován server schopný přijmout data z modulu XT75 ve formátu XML nebo CSV. Server je realizovaný jako PHP skript (dále míněný jako XML server) odpovídající na data poslaná přes HTTP spojení metodou POST. Jméno skriptu zpracovávajícího data je „xt75server.php“

Funkce XML serveru je následující. Server si nejprve ověří, zda-li přichází data z HTTP spojení používají metodu POST. Pokud přichází data pocházejí z požadavku POST, jsou stažena přes standardní vstup 'php://input' do proměnné. Následně server tyto data prohledá a zjistí, zda byly modulem XT75 data poslána ve formátu XML nebo CSV.

Podle formátu dat server pokračuje v dalším zpracování. V případě dat XML tyto data parsuje podle značek, v případě CSV je parsuje podle oddělovačů - čárek. V obou případech server vygeneruje shodný cílový soubor, který je dále zobrazován na www stránce. Tento soubor je datumově pojmenován, každý den má tedy svůj vlastní soubor. Pro příklad data ze dne 2.3.2010 jsou uložena v souboru s názvem '20100302\_temp.txt'. Zpracováním dat a vytvořením souboru s odpovídajícím datem práce XML serveru končí.

## 9. WWW rozhraní pro zpřístupnění naměřených hodnot

Pro zpřístupnění naměřených dat slouží webové rozhraní naprogramované v jazyku PHP. Hlavními částmi webového rozhraní jsou naměřená data, skript kalendáře a skript pro zobrazení naměřených dat v grafu.

Samotná naměřená data jsou zpracovávána s předstihem pomocí XML serveru vždy v době poslání dat na server. XML server tyto data nachystá do adresáře „data“ tak, že každý soubor je označen dnem, ve kterém probíhalo měření teplot. Se soubory následně pracuje skript kalendáře a skript pro zobrazení grafu.

Funkce kalendáře byla vytvořena přepsáním a počestěním skriptu PHP kalendáře poskytovaného zdarma, viz. literatura [8]. Skript kalendáře v průběhu svého zobrazování jednotlivých dní prohledává adresář dat a hledá soubor s názvem shodujícím se dne. Pokud takový soubor nalezne, zvýrazní datum v kalendáři a vytvoří nad ním symbolický link, který následně slouží pro zobrazení teplot tohoto dne v grafu.

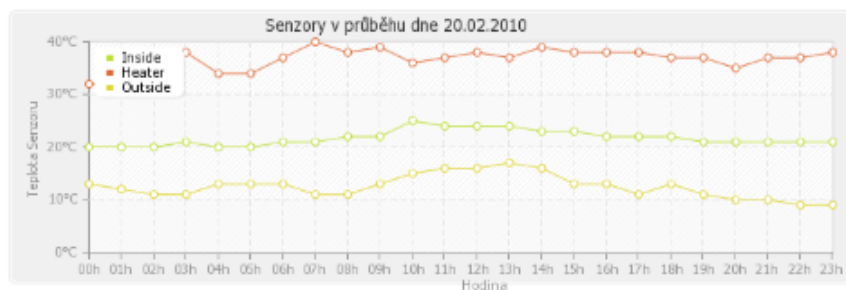
Skript pro zobrazení grafu byl přejat z PHP třídy poskytované zdarma pro vytváření grafů, viz. literatura [9]. I přes toto zvýhodnění bylo nutno pomocí PHP skriptu data správně přizpůsobit k předání této třídě pro jejich zobrazení. Data naměřených teplot jsou z www rozhraní přístupná pod menu Grafy hodnot.

Obsah webové stránky grafy hodnot je logicky rozdělen do tří částí. První část slouží k výběru dne pomocí kalendáře na obrázku 9.1, jehož průběh teplot se zobrazí v následující druhé části jako graf teplot na obrázku 9.2. Třetí část zobrazuje tři grafy aktuálně vybraného měsíce, každý graf pak ukazuje maximální a minimální naměřenou teplotu v průběhu měsíce na každém senzoru, viz. obrázek 9.3.

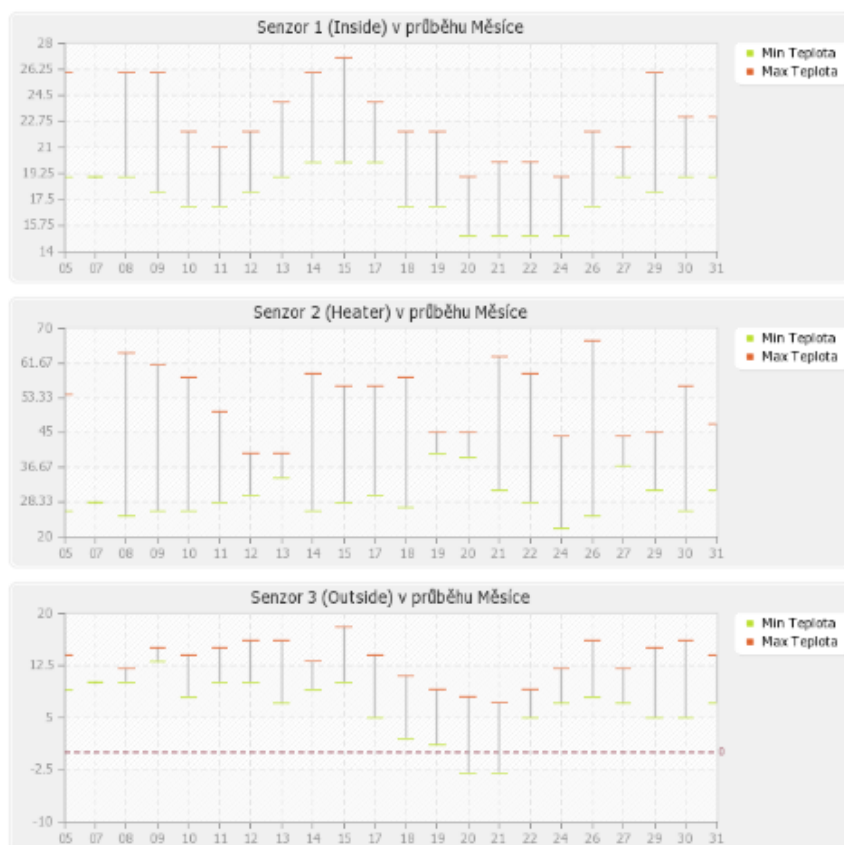
Leden      Únor - 2010      Březen

Po	Út	St	Čt	Pá	So	Ne
	01	02	03	04	05	06
07	08	09	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

Obr. 9.1: Kalendář pro výběr dne.



Obr. 9.2: Teploty v průběhu vybraného dne.

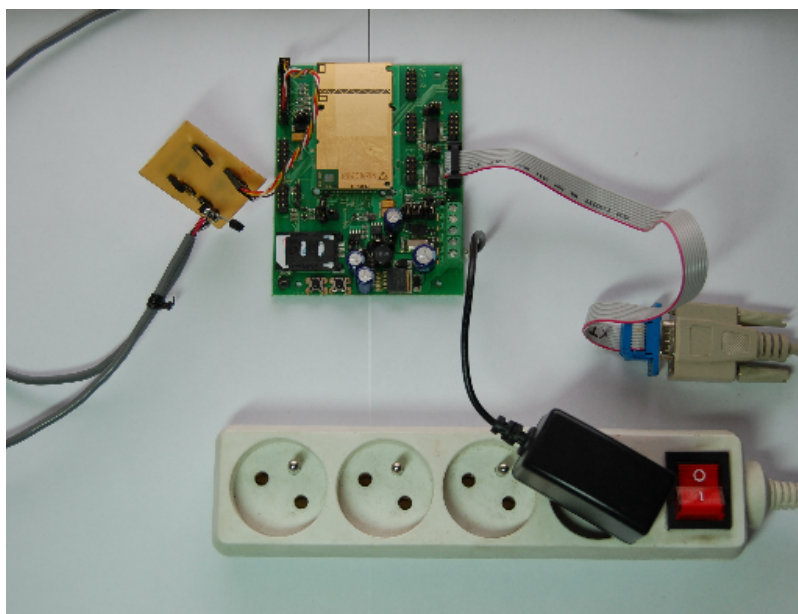


Obr. 9.3: Maximální a minimální teploty v průběhu měsíce.

## 10. Reálný provoz zařízení

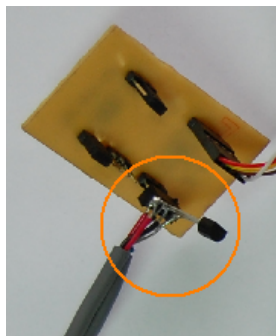
Navržené zařízení M2M senzorového modulu bylo také řádně otestováno. V průběhu testů vzniklo několik měření v různých dnech, která jsou všechna dostupná na webových stránkách. První data jsou dostupná od Prosince 2009. Samotné zařízení a senzory teplot byly umístěny v rámci možností v domácích podmínkách.

Na obrázku 10.1 je zobrazen modul XT75, základní deska M2M Universal Motherboard, SIM karta, 12V napájecí adaptér, připojení RS232 k vývojovému PC a propojení modulu XT75 přes M2M Universal Motherboard s externím modulem hardwarového mostu DS2482-100, ze kterého je následně vyvedena kabeláž typu CAT5 pro propojení teplotních senzorů. GSM anténa, která se obvykle připojuje k modulu XT75 není vyobrazena, neboť zařízení se v průběhu testů nacházelo v hustě osídlené zástavbě, a pracovalo spolehlivě i bez připojené antény.



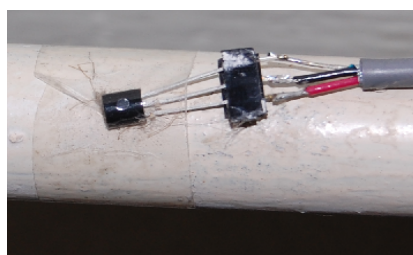
*Obr. 10.1: Modul XT75 v praxi.*

Konkrétní umístění senzorů je na následujících obrázcích. Senzor číslo jedna s popisem Inside vyobrazený na obrázku 10.2 je osazen uprostřed sběrnice, která je ve stejném místě připojena k modulu hardwarového mostu. Tento senzor má za úkol měřit teplotu okolí ve kterém je zařízení umístěno.



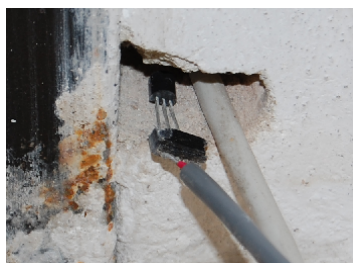
*Obr. 10.2: Senzor č.1 - Inside.*

Senzor číslo 2 s popisem Heater vyobrazený na obrázku 10.3 byl připevněn na přívodní trubku tělesa ústředního vytápění a měřil tak přibližnou teplotu přiváděnou do místnosti pomocí ústředního topení.



*Obr. 10.3: Senzor č.2 - Heater.*

Senzor číslo 3 s popisem Outside vyobrazený na obrázku 10.4 byl umístěn v otvoru vnější stěny, která spojuje vnitřní místnost s venkovním prostředím, a která je určena pro přívodní kabel internetového připojení od zařízení umístěného na střeše budovy. Vzhledem k tomu, že tento senzor nebyl umístěn přímo vně stěny, naměřené teploty neodpovídají přesně teplotě venkovní. To ale pro testovací účely nebylo na závadu, bylo záměrem mít pouze různé umístění senzorů tak, aby vzájemně měřily různou teplotu, čehož bylo dosaženo.



*Obr. 10.4: Senzor č.3 - Outside.*

## 11. Závěr

Tato práce ukazuje, že mobilní technologii GSM lze použít daleko širším způsobem než je poskytování hlasových nebo datových služeb koncovým zákazníkům. Je zde představena práce s GSM modulem firmy Cinterion XT75 a zejména možnost a způsob jak připojit k tomuto zařízení i jiné hardwarové prostředky než samo o sobě poskytuje, v našem případě tři teplotní senzory připojené ke sběrnici 1-Wire.

Původně návrh předpokládal připojení hardwarového mostu k modulu XT75 přes sběrnici I<sup>2</sup>C, kterou standardně poskytuje. V průběhu vývoje zařízení vyšlo najevo, že I<sup>2</sup>C sběrnice modulu XT75 plně nekopíruje standard I<sup>2</sup>C a není možno na jeho sběrnici použít příkaz opakovaného startu. Vzhledem k tomu, že hardwarový most tento příkaz vyžaduje ke své činnosti, byla nakonec zvolena možnost napsání protokolu I<sup>2</sup>C softwarově s použitím dvou vstupně/výstupních linek modulu XT75. Ačkoliv se toto řešení ukázalo jako funkční, přesto díky vlastnostem vstupně/výstupních linek modulu XT75 a to zejména k přístupu na tyto linky pomocí AT příkazů a časovému zpoždění, lze konstatovat, že softwarové I<sup>2</sup>C v modulu XT75 je příliš pomalé pro nějakou vážnější práci a vybudování rozsáhlé 1-Wire sběrnice s mnoha čidly.

V průběhu práce jsem zrealizoval nové myšlenky, například použití offset času a získání tak rychlejšího přístupu k obvodu RTC modulu XT75 než přes standardní rozhraní AT příkazem. Návrh by se dal dále i vylepšit, například pomalou komunikaci softwarové implementace I<sup>2</sup>C přes vstupně/výstupní linky modulu XT75 by bylo možno zrychlit použitím ovládání těchto linek přímo z programu J2ME. Další potenciál pro zrychlení komunikace se skrývá v samotném teplotním čidle. Aby mohla být zahájena konverze teploty, musí být každé čidlo teploty nejdříve adresováno a vykonán příkaz pro konverzi teploty. Použitá čidla ale podporují příkaz, kdy lze konverzi zahájit i bez adresace. To znamená, že je možno zahájit konverzi teploty na všech třech teplotních senzorech současně a následně z každého z nich pak jednotlivě získat naměřenou teplotu. Nicméně i po implementaci těchto rychlostních vylepšení, by bylo použití většího počtu senzorů na sběrnici 1-Wire stále omezené. Dále by bylo možno návrh rozšířit například o připojení jiných senzorů, například 1-Wire AD převodníku nebo vzdálené IO linky, vyžadovalo by to ovšem napsat v J2ME pro každý typ takového nového rozšíření obslužnou třídu. Jiným rozšířením by mohlo být například zpětné ovládání modulu ze vzdáleného místa přes SMS. Příkladem může být například změna výstupního formátu dat zasílaného na XML server nebo požadavek na okamžité změření teploty a její odeslání na XML server. Další rozšíření by mohlo být například přechod z vývojové desky na modul připravený v krabičce, vybavený bateriovým napájením a dostupný tak koncovému uživateli.



Mě osobně práce na tomto návrhu přinesla nový náhled do problematiky GSM a M2M aplikací. Do té doby jsem si nebyl schopen uvědomit jiné možnosti aplikace než GSM hlasových a datových služeb, jak jsou dnes převážně známy. Díky této práci jsem se blíže seznámil s modulem XT75 a na získaných znalostech a zkušenostech mohu dále budovat.

## 12. Literatura

[1] Maxim Integrated Products: Guidelines for Reliable Long Line 1-Wire® Networks, Application note 148: [www.maxim-ic.com/an148](http://www.maxim-ic.com/an148), Copyright © by Maxim Integrated Products, Sep 22, 2008

[2] Maxim Integrated Products: Determining the Recovery Time for Multiple-Slave 1-Wire Networks, Application note 3829: [www.maxim-ic.com/an3829](http://www.maxim-ic.com/an3829), Copyright © by Maxim Integrated Products, Jun 21, 2006

[3] Jean-Marc Irazabal, Steve Blozis: I<sup>2</sup>C MANUAL, Application note AN10216-01: [www.st.com/standardproducts/I2C](http://www.st.com/standardproducts/I2C), Philips Logic Product Group (NXP Semiconductors), March 24, 2003

[4] Philips Semiconductors: The I<sup>2</sup>C-bus and how to use it (including specifications), Philips Semiconductors (NXP Semiconductors), April, 1995

[5] Maxim Integrated Products: DS18S20 High-Precision 1-Wire Digital Thermometer, Datasheet, REV: 042208, Copyright © by Maxim Integrated Products, April, 2008

[6] Maxim Integrated Products: DS2482-100 Single-Channel 1-Wire Master, Datasheet, Rev 8; 11/09, Copyright © by Maxim Integrated Products, November, 2009

[7] Embedded Systems Academy, Inc.: I2C Driver in Psudocode, © Copyright Embedded Systems Academy 1999 – 2010, <http://www.esacademy.com/en/library/technical-articles-and-documents/miscellaneous/i2c-bus/miscellaneous-information/i2c-driver-in-psudocode.html>

[8] Geo Varghese: Simple PHP Calendar Script, © Copyright 2009 [www.seofreetools.net](http://www.seofreetools.net), <http://www.seofreetools.net/scripts/php-calendar/>

[9] Jean-Damien POGOLOTTI: pChart - a PHP class to build charts, <http://pchart.sourceforge.net>

[10] Maxim Integrated Products: How to Use the DS2482 I<sup>2</sup>C 1-Wire® Master, Application note 3684: [www.maxim-ic.com/an3684](http://www.maxim-ic.com/an3684), Copyright © by Maxim Integrated Products, Dec 17, 2008